

Jegyzőkönyv

Web Technológiák féléves feladat

Téma: PC Webshop

https://github.com/GaborCsicsely/C7H5VB_WebTech/tree/main

Csicsely Gábor

C7H5VB

2024.01.09

Feladat leírás:

A beadandóra, egy adatbázissal kezelt gépeket árusító webshopot csináltam, ahol meglévő gépek és saját magunknak összerakott gépek közül lehet vásárolni.

Az adatok tárolására főként SQL adatbázist használtam, de néhány funkció megbízható működéséhez ezt kombináltam localStorage-al. Ez éles környezetben nem biztonságos, de a beadandóhoz, tökéletes megoldást nyújt a bejelentkezés megjegyzéséhez, valamint a lekérdezések könnyítéséhez.

Az adatbázis létrehozásához MSSMS (Microsoft SQL Server Management Studio) volt használva.

Használt programozási nyelvek, IDE:

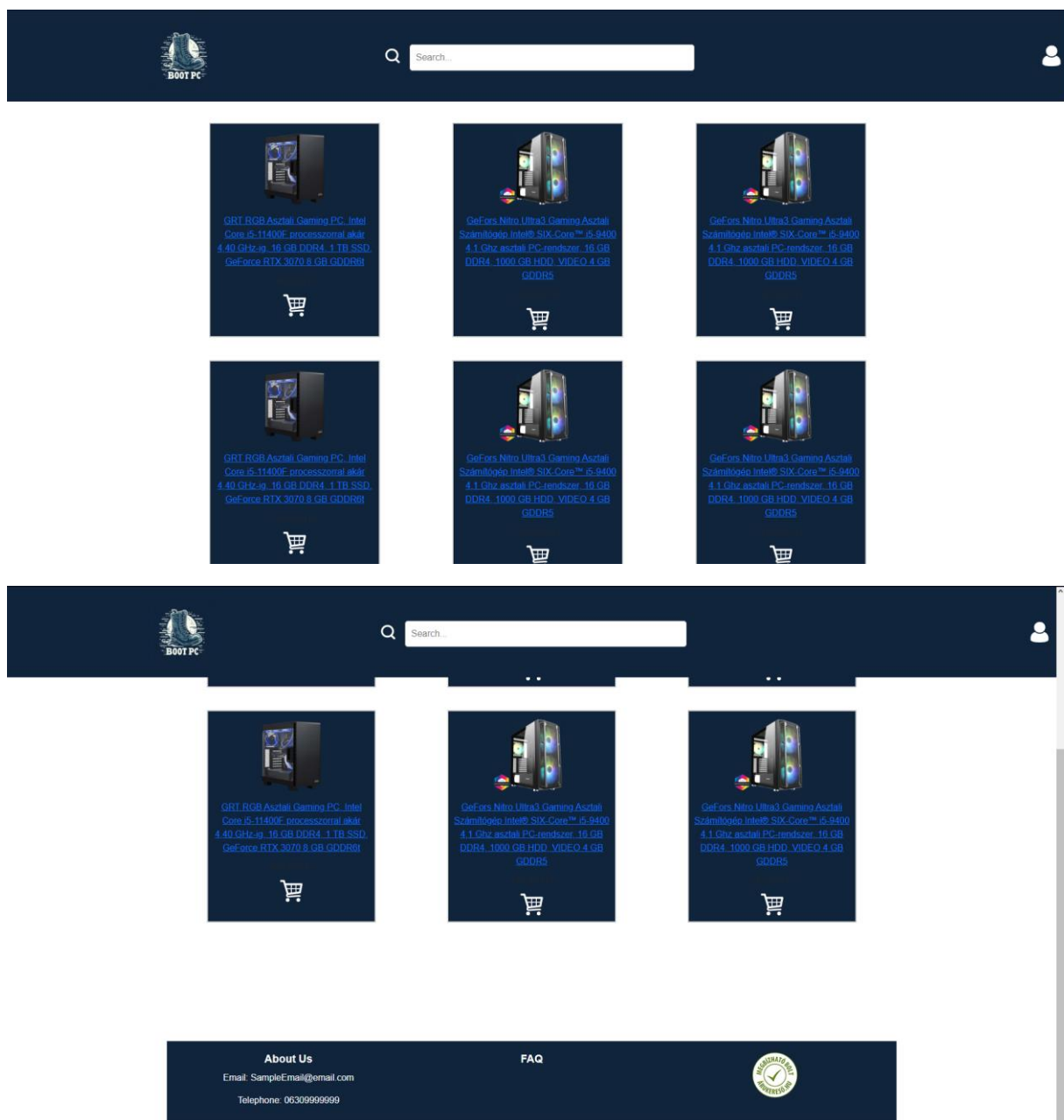
A Projekt maga az ASP.NET-en alapul és Visual Studio IDE-n lett megírva.

Az ASP.NET-et azért választottam, mert lehetővé tette az adatbázis használatát, és annak kezelését C# segítségével.

A weboldalak HTML-ben voltak megírva, ennek formázására pedig CSS-t használtam és legtöbb funkcióhoz pedig JavaScriptet.

A C# csak a DB kezelésére volt használva.

A Main Page, Header és Footer:

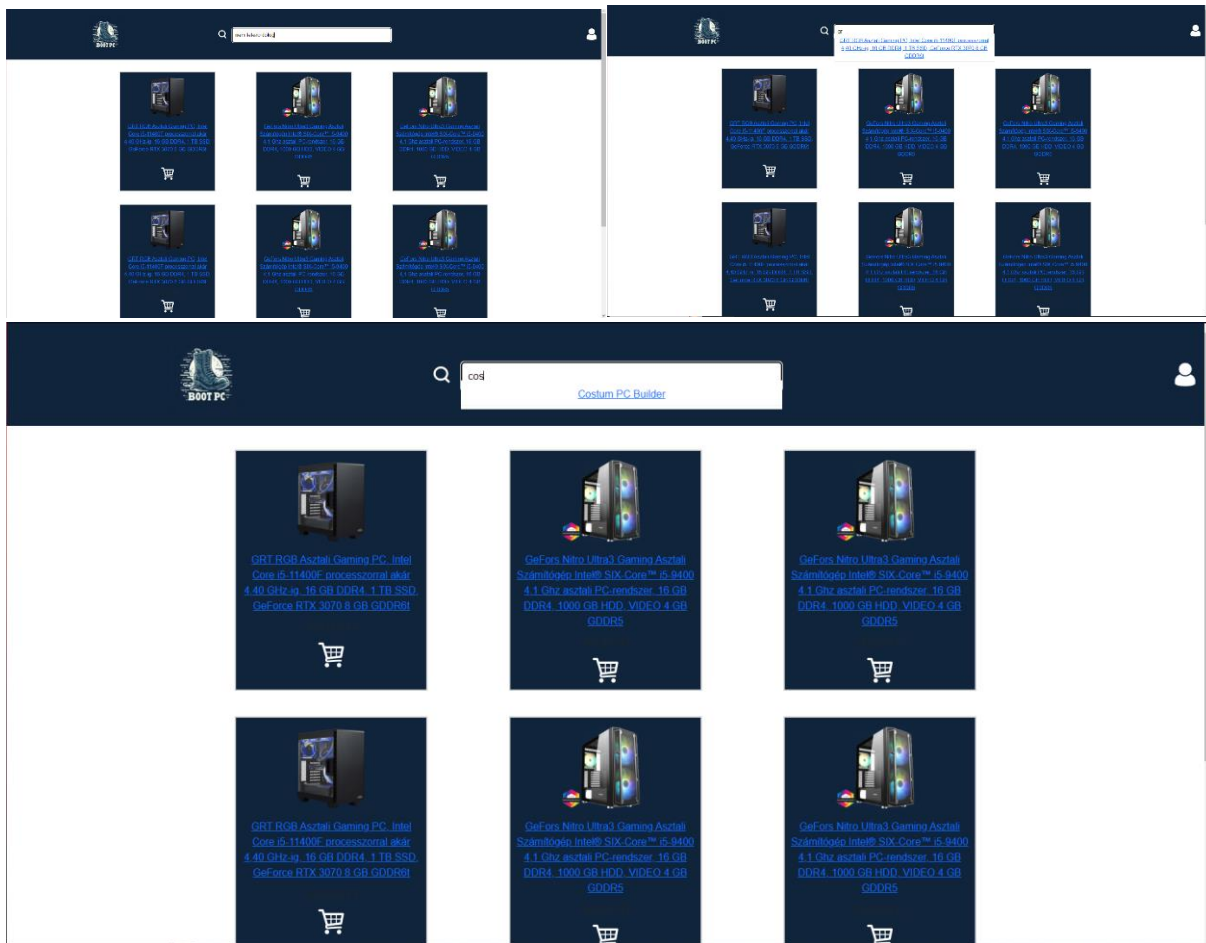


Ami az összes oldalon egyezik, a felül lévő sticky header, valamint a footer a weboldalak alján.

Ezekon található a search bar, a weboldal logó (és egyben a fő oldal linkje) és az Account Page/Login page.

A Fő oldal lényegében 2 gép linkjeit tartalmazza árakkal, képpel és címmel, viszont ezeket a tér kitöltése érdekében többszörösen hozzá adtam.

A Search Bar:



A Search bar lényegében egy olyan Input mező melyekhez hozzá van kapcsolva egy JS Function.

Elsőnek htmlPages tömbbe létrehozuk azokat az oldalakat amiket elérhetővé szeretnénk tenni keresésre(Én a costum PC builder és a Computer1 oldalt adtam hozzá)

Ezeknek elsőnek megadjuk az elérését, majd a keresési nevét.

Amikor elkezdünk írni bele a searchTerm változó segítségével keresünk a htmlPages-ben a filter függvénnnyel.

A talált elemekre létrehoz egy lista elemet, linkkel az elérhetőségével.

Computer1 és Computer2:



GeFors Nitro Ultra3 Gaming Asztali Számítógép Intel® SIX-Core™ i5-9400 4,1 Ghz asztali PC-rendszer, 16 GB DDR4, 1000 GB HDD, VIDEO 4 GB GDDR5

BUDGET

ÁLTALÁNOS TULAJDONSÁGOK

Konfiguráció típusa	Asztali számítógép
Használat	Gamer
Operációs rendszer	Free DOS
Csomag tartalma	1 x Asztali számítógép
Hosszúság	198 mm
Mélység	413 mm
Magasság	459 mm
Szín	Fekete

PROCESSZOR

Processzor gyártó	Intel®
-------------------	--------



GRT RGB Asztali Gaming PC

ÁLTALÁNOS TULAJDONSÁGOK

Konfiguráció típusa	Asztali számítógép
Használat	Gamer
Operációs rendszer	Free DOS
Csomag tartalma	1 x Asztali számítógép
Hosszúság	198 mm
Mélység	413 mm
Magasság	459 mm
Szín	Fekete

PROCESSZOR

Processzor gyártó	Intel®
Processzor típusa	i5
Processzor model	11400F
Processzormagok száma	6
Szállak száma	12
Architektúra	Rocket Lake
Alap frekvencia	2.6 GHz
Turbo Boost frekvencia	4.4 GHz
Cache	12 MB
Processzor gyártási technológia	14 nm
Integrált grafikus processzor	Nincs integrált grafikus processzor

ALAPLAP

Processzor foglalat	LGA 1200
Chipset	Intel H510
Alaplap csatlakozók	1 x PCI Express x16
Hátpanel portok	1 x 7.1 csatornás hang
Memória foglalatok száma	2



GRT RGB Asztali Gaming PC

ÁLTALÁNOS TULAJDONSÁGOK

Konfiguráció típusa	Asztali számítógép
Használat	Gamer
Operációs rendszer	Free DOS
Csomag tartalma	1 x Asztali számítógép
Hosszúság	198 mm
Mélység	413 mm
Magasság	459 mm
Szín	Fekete

PROCESSZOR

Processzor gyártó	Intel®
Processzor típusa	i5
Processzor model	11400F
Processzormagok száma	6
Szállak száma	12
Architektúra	Rocket Lake
Alap frekvencia	2.6 GHz
Turbo Boost frekvencia	4.4 GHz
Cache	12 MB
Processzor gyártási technológia	14 nm
Integrált grafikus processzor	Nincs integrált grafikus processzor

ALAPLAP

Processzor foglalat	LGA 1200
Chipset	Intel H510
Alaplap csatlakozók	1 x PCI Express x16
Hátpanel portok	1 x 7.1 csatornás hang
Memória foglalatok száma	2

Computer2 oldalon animált BUDGET felirat:


https://github.com/GaborCsicsely/C7H5VB_WebTech/blob/main/animation.mp4

Ezen az oldalon láthatunk egy táblázatos leírást a megvásárolható gépről, valamint a JS segítségével választhatunk a képek között az egerünkkel.

Ha letekerünk az Order gombbal megtudjuk rendelni a terméket.

Ezt az „addToCart()” functionnel oldottam meg, melyben a DB felé küldünk egy kérést a CheckoutController C# kód segítségével.

DESKTOP-E358PSH....ad - dbo.Checkout			
	Column Name	Data Type	Allow Nulls
▶	CheckoutId	int	<input type="checkbox"/>
	UserID	int	<input checked="" type="checkbox"/>
	Price	decimal(18, 0)	<input checked="" type="checkbox"/>
	DateOrdered	date	<input checked="" type="checkbox"/>
	Address	varchar(50)	<input checked="" type="checkbox"/>
	Type	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

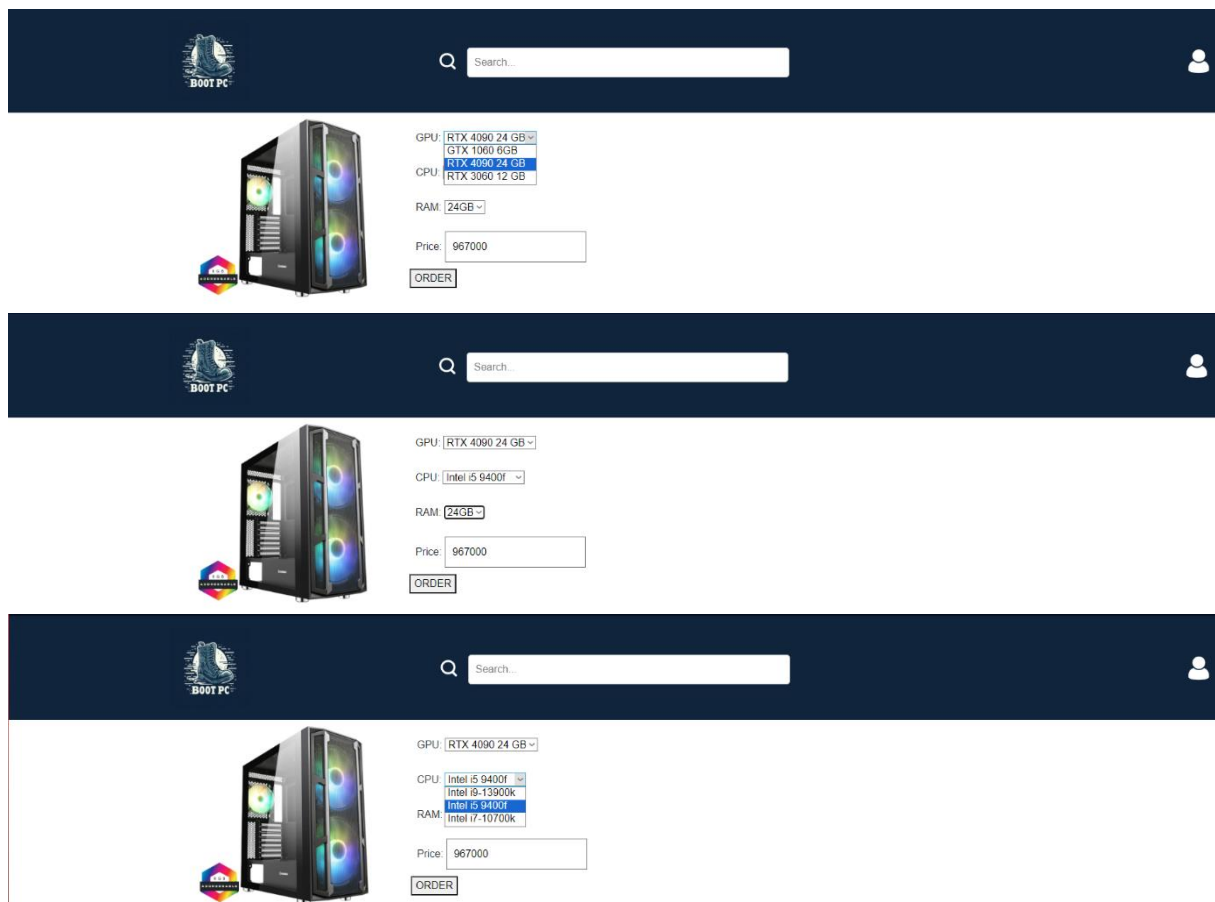


Rendelések

Checkout ID: 3
Price: 1,186,000 Ft
Date Ordered: 12/4/2023
Address: Address
Type: Custom Build: 213

Checkout ID: 4
Price: 967,000 Ft
Date Ordered: 1/10/2024
Address: Address
Type: Custom Build: 223

Costum PC Builder:



The image displays three sequential screenshots of a web-based PC configuration tool. Each screenshot shows a dark blue header with the 'BOOT PC' logo, a search bar, and a user icon. The main content area features a 3D rendering of a PC tower on the left and a list of configurable components on the right. The components are GPU, CPU, and RAM, each with a dropdown menu. A 'Price' field and an 'ORDER' button are also present. The first screenshot shows the initial state with default selections. The second screenshot shows the CPU dropdown menu open, displaying options like 'Intel i5 9400f'. The third screenshot shows the RAM dropdown menu open, displaying options like 'Intel i7-10700k'.

GPU: RTX 4090 24 GB
GTX 1060 6GB
RTX 4090 24 GB
CPU: RTX 3090 12 GB
RAM: 24GB
Price: 967000
ORDER

GPU: RTX 4090 24 GB
CPU: Intel i5 9400f
RAM: 24GB
Price: 967000
ORDER

GPU: RTX 4090 24 GB
CPU: Intel i5 9400f
Intel i6-13900k
Intel i5-9400f
RAM: Intel i7-10700k
Price: 967000
ORDER

Ennek a felépítéséhez a html forms-át használtam, select-el.

Az árát a kiválasztott elemek alapján egy JS function kiszámolja. Ez úgy lett megoldva h minden opciónak adtam egy értéket, amiket összead és kiszámol.

Amikor megrendeljük a gépet, szerettem volna, hogy az adatbázisban/felhasználói oldalon látható legyen, hogy milyen opciókat választott a felhasználó. Ezt is egy JS function segítségével oldottam meg, melyeben a kiválasztott elemeknek egy bizonyos számot adtam meg 1-3ig.

Így az adatbázisba például, ha az 1. GPU-t 3. CPU-t és 2. RAMot választotta, a kódja Custom Build: 132 lesz.

Login/Register:

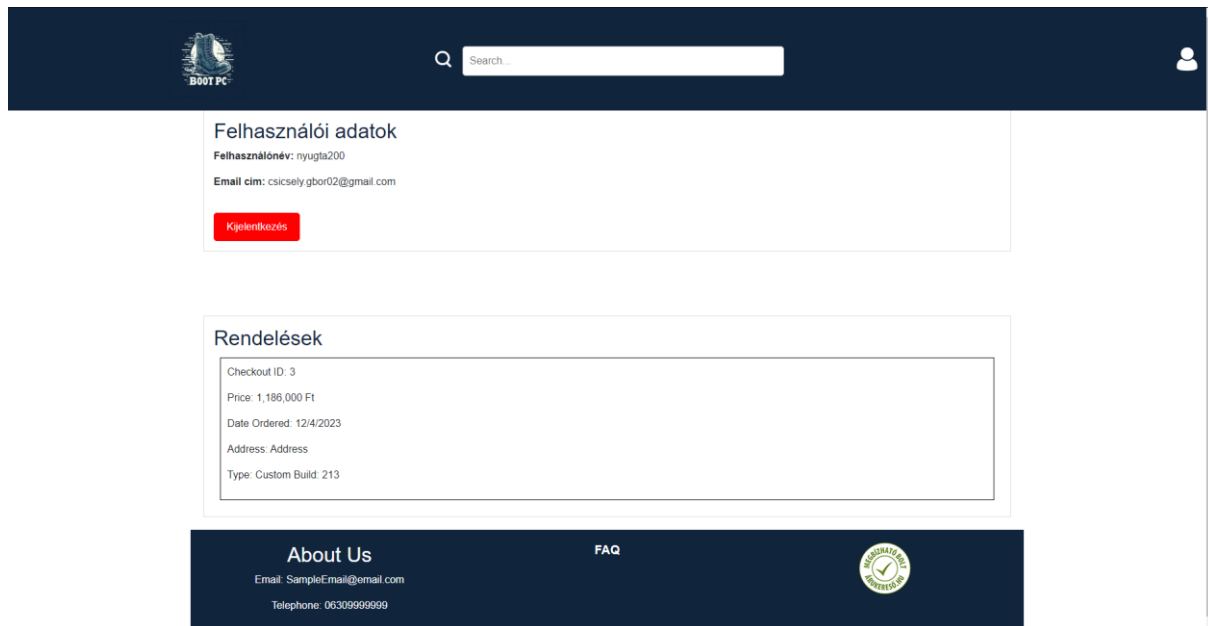
The image displays three screenshots of a web application's login/register page. The top two screenshots show the initial form with fields for Email, Username, Password, and Confirm Password. The bottom screenshot shows the form after a successful login, with the Username field filled with 'nyugta200' and the Password field masked with '***'. A notification box at the top of the bottom screenshot says 'localhost5000 says: Bejelentkezve a sikeresen.'

Amikor rákattintunk a Headerben az account ikonra, az oldal ellenőrzi, hogy éppen bevan-e jelentkezve a felhasználó LocalStorage-ban (`userLoggedIn = True/False`) és ez alapján visz vagy a Login, vagy az Account oldalra.

Az oldal itt is Formsot használ, melyben az összes mező required.

Ez után, ha kitöltjük ellenőrzi az DB-ben lévő adatokat és az alapján vagy hiba üzenetet ad és nem engedi a bejelentkezést/regisztrációt (Ez lehet azért mert már foglalt az email, vagy mert helytelenek az adatok), vagy engedélyezi és továbbít minket az account page-re, és `userLoggedIn = True`-t, más fontosabb adatokat felvisz a LocalStorage-ba.

Account Page:



Ezen az oldalon találhatjuk meg a leadott rendeléseket, valamint a felhasználói adatainkat és a kijelentkezés gombot.

A felhasználói adatokat egyszerűen kivesszük LocalStorageból, ez kevésbé biztonságos, mintha SQL hívással szereztük volna meg őket, viszont gyorsabb és sokkal megbízhatóbban működik.

A kijelentkezés gombunk, annyit csinál, hogy törli a LocalStorageból a felhasználói adatokat, valamint a userLoggedIn változót aztán visszadob a főoldalra.

A rendelések viszont DB-ből vannak lekérdezve, a bejelentkezett felhasználó UserID-ja alapján

FAQ oldal:

The image shows two side-by-side screenshots of a web application's FAQ page, with the right side overlaid by the Redux DevTools application state viewer.

Left Screenshot (FAQ Page):

- Header: BMTFC logo, search bar, and user icon.
- Section: Frequently Asked Questions (FAQ)
- Text: "Kérdezze: Összegyűjtött kérdések" and "Ha nem találja meg a kérdését, kérjük, írja meg a kérdését. Válaszra várunk a kapcsolattal."
- Form: Name, Email, and Message input fields, a "Send" button, and a checkbox for "Rendelést kapcsolódással küldök".

Right Screenshot (Redux DevTools):

- Application: http://localhost:5000
- Storage: Local storage, http://localhost:5000, IndexedDB, Web SQL, Cookies, Private state tokens, Interest groups, Shared storage, Cache storage.
- Background services: Back/forward cache, Background fetch, Background sync, Bounce tracking mitigations, Notifications, Payment handler, Periodic background sync, Push messaging, Reporting API.
- Preloading: Speculation rules, Preloads, This page.
- Frames: top.
- State: A table showing the state of the application, including a "formData" object with fields like "username", "UserID", "email", "debug", and "userLoggedIn".

Key	Value
formData	{ "userID": "2", "name": "Csicsely Gábor", "email": "csicsely.gbor02@gmail.com", "message": "teszt" }
username	nyugta200
UserID	2
email	csicsely.gbor02@gmail.com
debug	honeycore-sdk*
userLoggedIn	true

Below the table, a log entry is visible: `{ "userID": "2", "name": "Csicsely Gábor", "email": "csicsely.gbor02@gmail.com", "message": "teszt", ... }`

Itt is listát használva van egy rövid gyakori kérdések, rá válaszok.

Alatta van egy form, melyben a felhasználó üzenetet küldhet felénk, amit utána LocalStorage-ban tárolunk.

Itt belehet pipálni, hogy a kérdés rendeléssel kapcsolatos-e, és ha igen akkor megjelenik egy dátum választó, melyben a rendelés dátumát kell megadni. Ha a felhasználó bevan jelentkezve UserID-val együtt van mentve az üzenet.

Az adatbázis:

DESKTOP-E358PSH (SQL Server 16.0.10)

Databases

System Databases

Database Snapshots

CSBead

Database Diagrams

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.Checkout

dbo.OrderItems

dbo.Products

dbo.User

Dropped Ledger Tables

Views

External Resources

Synonyms

Programmability

Query Store

Service Broker

Storage

Security

	Column Name	Data Type	Allow Nulls
PK	UserID	int	<input type="checkbox"/>
	Email	varchar(50)	<input checked="" type="checkbox"/>
	Username	varchar(50)	<input checked="" type="checkbox"/>
	Password	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

	Column Name	Data Type	Allow Nulls
PK	CheckoutId	int	<input type="checkbox"/>
	UserID	int	<input checked="" type="checkbox"/>
	Price	decimal(18, 0)	<input checked="" type="checkbox"/>
	DateOrdered	date	<input checked="" type="checkbox"/>
	Address	varchar(50)	<input checked="" type="checkbox"/>
	Type	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

	Column Name	Data Type	Allow Nulls
	OrderItemID	int	<input checked="" type="checkbox"/>
	OrderID	int	<input checked="" type="checkbox"/>
	ProductID	int	<input checked="" type="checkbox"/>
	Quantity	numeric(18, 0)	<input checked="" type="checkbox"/>
	Price	money	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

	CheckoutId	UserID	Price	DateOrdered	Address	Type
1	3	2	1186000	2023-12-04	Address	Custom Build: 213
2	4	2	967000	2024-01-10	Address	Custom Build: 223

	UserID	Email	Username	Password
1	1	aasaa@das.com	aasaa	aasaa
2	2	csicsely.gbor02@gmail.com	nyugta200	asd
3	3	tester@tester.test	tester	tester
4	4	asd@asd.com	asd	asd
5	5	ujregister@register.com	register	register

Megjegyzés:

Sajnos a Visual Studio sokszor véletlenszerűen nem akarta frissíteni a változtatásokat a JS/CSS kódjaimban, ezért 4 különböző .js és 3 különböző .css fájlt tartalmaz a projekt, ezek között kisebb, nagyobb különbségek vannak, emellett néhány weboldal csak a saját maga a html-oldalba írt js kódját használja.

A DB működéséhez C# segítségével DBContextet kellett létrehozni melyben a táblákat megneveztük/létrehoztuk, emellett az összes tábláról kellett készíteni egy modellt amiben a tábla felépítését hoztuk létre.